

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



KOREAN PATENT ABSTRACTS(KR)

Document Code:A

(11) Publication No.1020010076707 (43) Publication.Date. 20010816

(21) Application No.1020000004049 (22) Application Date. 20000127

(51) IPC Code:

H04N 7/24

(71) Applicant:

SAMSUNG ELECTRONICS CO., LTD.

(72) Inventor:

CHOI, SEONG GYU

(30) Priority:

(54) Title of Invention

MOTION ESTIMATOR

Representative drawing



(57) Abstract:

PURPOSE: A motion estimator is provided to perform motion estimation at a high speed in the course of compressing a moving picture.

CONSTITUTION: A motion estimator is provided with the first memory(200) divided into a memory(202) for a priory searched region and a memory (204) for an on-processing region, an SDA(Sum of Absolute difference) calculation part (210:PE), an interpolation part (212), an adder(220) and a buffer(222). The motion estimator also has 25 shift register(230), an SDA comparing and motion vector generating part(240), a 22 shift register (250), a DMA(Direct Memory

Access) control part(260), a part(270) for controlling a motion estimation, and the second memory(280). The DMA control part(260) controls and classifies data to be stored in the priory searched region and in the on-processing region in the first memory(200). The DMA control part(260) also controls the data to be provided to the SDA calculation part(210) based on the processing of the part(270) for controlling

the motion estimation.

COPYRIGHT 2001 KIPO

if display of image is failed, press (F5)

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl. ⁷ H04N 7/24	(11) 공개번호 (43) 공개일자	특2001-0076707 2001년08월16일
(21) 출원번호 (22) 출원일자	10-2000-0004049 2000년01월27일	
(71) 출원인	삼성전자 주식회사	
(72) 발명자	경기 수원시 팔달구 매탄3동 416 최성규	
(74) 대리인	경기도광명시하안동하안주공아파트110동702호 이영필, 권석홍, 이상용	
심사청구 : 없음		
(54) 움직임 추정기		

요약

동영상의 압축 과정에서 사용되는 움직임 추정을 고속으로 수행할 수 있는 움직임 추정기가 개시된다. 본 움직임 추정기는 이전 탐색 영역의 데이터를 라인별로 저장하는 복수 개의 메모리 영역을 구비하는 이전 탐색 영역 메모리와 현재 처리 영역의 데이터를 라인별로 저장하는 현재 처리 영역 메모리를 구비하는 제1 메모리부와, 복수 개의 메모리 영역으로부터 순환적으로 출력되는 이전 탐색 영역의 데이터를 시프트시켜 출력하는 이전 탐색 영역 데이터 공급부와, 이전 탐색 영역 데이터 공급부로부터 출력된 이전 탐색 영역의 데이터를 선택하는 제1 스위칭부와, 현재 처리 영역 메모리로부터 출력되는 현재 처리 영역의 데이터를 행별로 선택하는 제2 스위칭부와, 선택된 이전 처리 영역의 데이터와 현재 처리 영역 데이터의 절대차합(sum of absolute difference)을 계산하는 복수 개의 절대차합 계산부가 병렬로 접속된 계산부, 및 계산부의 일단의 절대차합 계산부들로부터 출력된 절대차합 데이터를 입력하여 시프트를 수행하고 시프트된 데이터와 다음 단의 절대차합 계산부로부터 출력된 SAD(절대차합) 값을 합하여 부분 SAD 값을 출력하는 시프트 레지스터부를 포함하는 것을 특징으로 한다. 본 움직임 추정기는 처리 속도가 빠르고, 특히, 반화소 단위의 탐색을 빠르고 효율적으로 수행할 수 있다.

대표도

도3a

명세서

도면의 간단한 설명

- 도 1은 종래의 계층적 움직임 추정 방법을 설명하기 위한 도면이다.
- 도 2는 본 발명의 실시예에 따른 움직임 추정기의 구조를 도시한 블록도이다.
- 도 3은 도 2의 움직임 추정기의 제1 메모리, 부분 SAD 계산부, 및 보관부를 상세히 도시한 도면이다.
- 도 4는 이전 탐색 영역의 데이터가 메모리에 배치되는 순서를 설명하기 위한 도면이다.
- 도 5는 4×4 픽셀로 이루어지는 현재 처리 블록의 데이터가 도 3의 계산부내의 레지스터들에 배치되는 순서를 나타낸 도면이다.
- 도 6은 도 2의 움직임 추정기에서의 데이터 흐름을 설명하기 위한 도면이다.
- 도 7은 본 발명에 따른 움직임 추정기에서 적용되는 반화소 단위의 탐색 과정을 설명하기 위한 도면이다.
- 도 8은 도 2에 도시한 움직임 추정기의 보관부의 일례를 나타낸 블록도이다.
- 도 9는 도 2의 본 발명에 따른 움직임 추정기에서 반화소 단위의 탐색을 적용하였을 경우의 움직임 추정기에서의 데이터 흐름을 설명하기 위한 도면이다.
- 도 10은 본 발명의 실시예에 따른 움직임 추정기에서 적용되는 계층적 탐색에 사용된 중간 계층 및 하위 계층에서의 탐색 영역을 설명하기 위한 도면이다.
- 도 11은 도 2에 도시한 움직임 추정기에서 주변 매크로 블록의 상관관계를 이용하여 초기 탐색점을 하나 더 선정하기 위한 22 시프트 레지스터의 구성예를 나타낸 도면이다.
- 도 12는 이전 탐색 영역의 데이터를 수직 방향으로 4:1 데시메이션(decimation)하는 과정을 설명하기 위한 도면이다.
- 도 13은 상위 계층에서의 데이터 처리 흐름을 설명하기 위한 도면이다.

- 도 14는 이전 탐색 영역의 데이터를 수직 방향으로 2:1 데시메이션(decimation)하는 과정을 설명하기 위한 도면이다.
- 도 15는 중간 계층에서의 데이터 처리를 설명하기 위한 도면이다.
- 도 16은 중간 계층에서의 데이터 처리 흐름을 설명하기 위한 도면이다.
- 도 17은 하위 계층에서의 데이터 처리를 설명하기 위한 도면이다.
- 도 18은 하위 계층에서의 데이터 처리 흐름을 설명하기 위한 도면이다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 움직임 추정기에 관한 것으로, 더 상세하게는 동영상 압축 과정에서 사용되는 움직임 추정을 고속으로 수행할 수 있는 움직임 추정기에 관한 것이다.

움직임 추정은, 비디오 코더에서, 입력되는 비디오 데이터의 시간적인 중복을 제거함으로써 압축 효율을 높이기 위하여 사용된다. 대부분의 일반적인 동영상 압축 표준들은 블록 기반의 코더이고, 이러한 블록 기반의 코더에서는 블록 정합 알고리즘(block matching algorithm)이 사용된다. 블록 정합 알고리즘은 블록 단위로 움직임 추정을 수행하는 알고리즘을 일컫는다. 블록 정합 알고리즘에 따르면, 이전 프레임의 탐색 영역 내에 위치하는 후보 블록들 중에서 현재 프레임의 블록과 가장 유사한 블록을 찾는다. 이때, 얻어지는 현재 프레임의 처리블록과 이전 프레임의 가장 유사한 후보 블록 사이의 상대적인 위치의 차이는 움직임 벡터(motion vector: MV)라 불리운다. 움직임 추정을 위해서는 SAD(sum of absolute differences) 값을 계산하는 것이 일반적이다. 16×16 블록을 단위로 하여 움직임 추정을 수행할 경우, SAD 값과 움직임벡터(MV)는 각각,

$$SAD(u,v) = \sum_i \sum_j |C(i,j) - P(i+u, j+v)|$$

$$MV = \arg \min_{u,v} SAD(u,v)$$

과 같이 계산된다. 여기서, $C(i,j)$ 는 현재 프레임의 블록 데이터, $P(i+u, j+v)$ 는 이전 프레임의 후보 블록의 데이터, (u,v) 는 움직임 벡터 후보를 나타낸다.

이와같은 계산을 기초로 하여 실제 움직임 추정기를 설계할 때는, 움직임 추정의 성능과 연산량이 고려되어야 한다. 움직임 추정에 있어서 연산량의 대부분은 SAD 값을 계산하는 것이 차지한다. 즉, 하나의 SAD 값을 계산하기 위해서는 뺄셈, 절대값, 및 누적 연산을 수행하여야 하고, 특히, 이러한 계산량은 처리 블록의 크기에 비례한다. 따라서, 종래의 움직임 추정 방법에 따르면, 연산량이 많아 움직임 추정을 위하여 시간이 많이 소요된다는 문제점이 있다.

상기 문제점을 해결하기 위한 다른 종래의 움직임 추정 방법으로써 계층적 움직임 추정 방법이 제안되었다. 도 1에는 상기 계층적 움직임 추정 방법을 설명하기 위한 도면을 나타내었다. 도 1에 나타난 계층적 움직임 추정 방법에 따르면, 하위계층의 이미지를 서브 샘플링함에 의하여 중간 계층, 및 상위 계층의 서브 샘플링된 이미지를 생성한다. 다음으로, 서브 샘플링된 상위 계층의 이미지에 대하여 SAD 값을 계산함으로써 중간 계층의 이미지에 대한 초기 탐색 영역 후보를 선택하고, 주변 매크로 블록과의 상관 관계를 사용하여 탐색 영역 후보를 더 선택한다. 이제, 선택된 초기 탐색 영역 후보들에 대하여 SAD 값을 계산함으로써 하위 계층의 이미지에 대한 그 SAD 값이 최소가 되는 지점의 최종 움직임 벡터(MV)를 구한다.

이러한 방법은 전역 탐색 방법에 비하여 성능 저하는 극히 적고, 처리 속도는 빠르다. 하지만, 보다 고속으로 움직임 추정을 수행하고자 하는 요구가 여전히 존재한다.

발명이 이루고자하는 기술적 과제

본 발명이 이루고자 하는 기술적 과제는 고속으로 움직임 추정을 수행하는 움직임 추정기를 제공하는 것이다.

발명의 구성 및 작용

상기 과제를 이루기 위하여 본 발명에 따른 움직임 추정기는 영상 데이터 프레임을 압축하기 위하여 움직임 추정을 수행하는 움직임 추정기에 있어서, 이전 탐색 영역의 데이터를 라인별로 저장하는 복수 개의 메모리 영역을 구비하는 이전 탐색 영역 메모리와, 현재 처리 영역의 데이터를 라인별로 저장하는 현재 처리 영역 메모리를 구비하는 제1 메모리부; 복수 개의 메모리 영역으로부터 순환적으로 출력되는 이전 탐색 영역의 데이터를 시프트시켜 출력하는 이전 탐색 영역 데이터 공급부; 이전 탐색 영역 데이터 공급부로부터 출력된 이전 탐색 영역의 데이터를 선택하는 제1 스위칭부; 현재 처리 영역 메모리로부터 출력되는 현재 처리 영역의 데이터를 행별로 선택하는 제2 스위칭부; 선택된 이전 처리 영역의 데이터와 현재 처리 영역 데이터의 절대차합(sum of absolute difference)을 계산하는 복수 개의 절대차합 계산부

가 병렬로 접속된 계산부; 및 계산부의 일단의 절대차합 계산부들로부터 출력된 절대차합 데이터를 입력하여 시프트를 수행하고 시프트된 데이터와 다음 단의 절대차합 계산부로부터 출력된 SAD(절대차합) 값을 합하여 부분 SAD 값을 출력하는 시프트 레지스터부;를 포함하는 것을 특징으로 한다.

또한, 상기 계산부는 탐색 영역의 크기를 p 라 할 때 p 개의 절대차합 계산부가 병렬로 접속되고, 상기 시프트레지스터부는 $2p + 1$ 시프트레지스터를 구비하는 것이 바람직하다.

또한, 상기 움직임 추정기는 입력 데이터를 버퍼링하는 버퍼; 버퍼로부터 출력된 데이터를 25 시프트하는 25 시프트레지스터; 부분 SAD 값과 25 시프트 레지스터로부터 출력된 데이터를 가산하여 SAD 값을 출력하는 가산기; 가산기로부터 출력된 SAD 값을 사용하여 움직임 벡터를 생성하는 SAD 비교 및 움직임 벡터 생성부; 및 움직임 벡터값을 20 시프트시켜 SAD 비교 및 움직임 벡터 생성부로 출력하는 20 시프트레지스터;를 더 포함하는 것이 바람직하다.

또한, 상기 움직임 추정기는 3 계층으로 계층적 움직임 추정을 수행할 때 중간 계층에서 주변 매크로 블록이 가지는 움직임 벡터의 상관관계를 이용하여 초기 탐색점을 추가로 얻기 위하여 22 시프트 동작을 수행하는 22 시프트레지스터;를 더 포함하는 것이 바람직하다.

또한, 상기 움직임 추정기는 움직임 추정을 수행하기 위하여 현재 데이터와 이전 탐색 영역 데이터를 구분하여 제1 메모리에 저장되도록 제어하고, 움직임 추정 제어부의 처리 상황에 따라 부분 SAD 계산부로부터 데이터를 공급하도록 제어하는 DMA 제어부; DMA 제어부로부터 출력된 제어신호에 응답하여 움직임 추정 동작을 제어하는 움직임 추정 제어부; 및 계산된 부분 SAD 값을 저장하는 제2 메모리부;를 더 포함하는 것이 바람직하다.

또한, 상기 제1 메모리는, 세 영역의 이전 탐색 영역 메모리와, 현재 처리 영역 메모리로 구분되고, 상기 계산부는, 이전 탐색 영역의 데이터와 현재 처리 영역의 데이터를 사용하여 절대차합을 계산하는 제1 절대차합 계산부; 제2 절대차합 계산부; 제3 절대차합 계산부; 및 제4 절대차합 계산부;를 구비하며, 상기 시프트 레지스터부는, 상기 제1 절대차합 계산부로부터 출력된 절대차합 데이터를 9 시프트시켜 출력하는 제1 9시프트레지스터; 상기 제1 9시프트레지스터의 출력 데이터와 상기 제2 절대차합 계산부의 출력 데이터를 합산하는 제1 가산기; 상기 제2 절대차합 계산부로부터 출력된 절대차합 데이터를 9 시프트시켜 출력하는 제2 9시프트레지스터; 상기 제2 9시프트레지스터의 출력 데이터와 상기 제3 절대차합 계산부의 출력 데이터를 합산하는 제2 가산기; 제3 계산부로부터 출력된 절대차합 데이터를 9 시프트시켜 출력하는 제3 9시프트레지스터; 상기 제3 9시프트레지스터의 출력 데이터와 상기 제4 절대차합 계산부의 출력 데이터를 합산하는 제3 가산기;를 구비하는 것이 바람직하다.

또한, 상기 제1 절대차합 계산부는 레지스터들(CR0, CR1, CR2, CR3)과, 일 입력단은 상기 레지스터들(CR0, CR1, CR2, CR3)의 출력단에 접속되고 타 입력단은 스위치들에 접속된 감산기들(SUB)을 구비하고, 상기 제2 절대차합 계산부는 레지스터들(CR4, CR5, CR6, CR7)과, 일 입력단은 상기 레지스터들(CR4, CR5, CR6, CR7)의 출력단에 접속되고 타 입력단은 스위치들에 접속된 감산기들(SUB)을 구비하며, 상기 제3 절대차합 계산부는 레지스터들(CR8, CR9, CR10, CR11)과, 일 입력단은 상기 레지스터들(CR8, CR9, CR10, CR11)의 출력단에 접속되고 타 입력단은 스위치들에 접속된 감산기들(SUB)을 구비하고, 상기 제4 절대차합 계산부는 레지스터들(CR12, CR13, CR14, CR15)과, 일 입력단은 상기 레지스터들(CR12, CR13, CR14, CR15)의 출력단에 접속되고 타 입력단은 스위치들에 접속된 감산기들(SUB)을 구비하는 것이 바람직하다.

또한, 상기 움직임 추정기는 이전 탐색 영역의 데이터를 보관하여 보관된 데이터를 출력하는 보관부;를 더 포함하는 것이 바람직하다.

또한, 상기 움직임 추정기는 이전 탐색 영역에 해당하는 데이터와 현재 처리 블록에 해당하는 데이터를 구분하는 동작을 수행하고, 처리 상황에 따라 상기 이전 탐색 영역 데이터 공급부와 상기 계산부로 데이터를 공급하도록 제어하는 DMA 제어부를 더 포함하는 것이 바람직하다.

이하 첨부된 도면들을 참조하여 본 발명의 바람직한 실시예들을 상세히 설명하기로 한다.

도 2에는 본 발명의 실시예에 따른 움직임 추정기의 구조를 블록도로써 도시하였다. 본 실시예의 움직임 추정기는 3 계층의 계층적 움직임 추정 방법을 기초로 한다. 하위 계층에서는 16×16 블록, 중간 계층에서는 8×8 블록, 및 상위 계층에서는 4×4 블록에 대하여 SAD 계산이 수행된다. 즉, 중간 계층은 하위 계층을 1/2 데시메이션한 계층이고, 상위 계층은 중간 계층을 1/2 데시메이션한 계층이다. 또한, 상위 계층에서는 ± 4 픽셀 단위의 탐색이 수행되고, 중간 계층 및 하위 계층에서는 ± 2 픽셀단위의 탐색이 수행된다. 또한, 본 실시예의 움직임 추정기는 하위 계층의 정화소 단위의 추정이 완료된 다음에 상기 하위 계층에 대하여 반화소 단위의 추정을 사용한다. 또한, 본 실시예에서는 본 출원인에 의하여 출원되고 본 명세서에 참조로써 통합되는 대한민국 특허출원 98-6718 호에서 개시된 개선된 움직임 추정 방법을 예로써 사용한다. 상기 특허출원에 개시된 움직임 추정 방법에 따르면, 중간 계층에서 선택된 매크로 블록 후보에 하위 계층에서 주변 매크로 블록들과의 상관 관계를 이용하여 적어도 하나 이상의 매크로 블록 후보를 추가로 선택한다.

도 2를 참조하면, 본 발명의 실시예에 따른 움직임 추정기는 제1 메모리(200), 부분 SAD 계산부(210: PE), 보관부(212), 가산기(220), 및 버퍼(222)를 구비한다. 또한, 상기 움직임 추정기는 25 시프트레지스터(230), SAD 비교 및 움직임 벡터 생성부(240), 22 시프트레지스터(250), DMA 제어부(260), 움직임 추정 제어부(270), 및 제2 메모리(280)를 구비한다. 제1 메모리는 이전 탐색 영역 메모리(202)와 현재 처리 영역 메모리(204)로 구분된다.

동작을 설명하면, DMA 제어부(260)는 움직임 추정을 수행하기 위하여 현재 데이터와 이전 탐색 영역 데이터를 구분하여 제1 메모리(200)에 저장되도록 제어한다. 또한, DMA 제어부(260)는 움직임 추정 제어부(270)의 처리 상황에 따라 부분 SAD 계산부(210: PE)로 데이터를 공급하도록 제어한다.

부분 SAD 계산부(210: PE)는 현재 처리 블록 데이터와 이전 탐색 영역 데이터를 입력하여 부분 SAD 값을 계산한다. 또한, 상기 움직임 추정기는 22 시프트 레지스터(250)를 구비함으로써 3 계층으로 계층적 움

직임 추정을 수행할 때 중간 계층에서 주변 매크로 블록이 가지는 움직임 벡터의 상관관계를 이용하여 초기 탐색점을 얻는다.

도 3에는 도 2의 움직임 추정기의 제1 메모리(200), 부분 SAD 계산부(210), 및 보간부(212)를 상세히 도시하였다. 도 3을 참조하면, 제1 메모리(200)는 세 영역의 이전 탐색 영역 메모리(이전 탐색영역 메모리_A, 이전 탐색영역 메모리_B, 이전 탐색영역 메모리_C)와, 현재 처리 영역 메모리로 구분된다. 또한, 부분 SAD 계산부(210)는 이전 탐색 영역 데이터 공급부(30), 계산부(32), 및 시프트레지스터부(34)를 구비한다. 또한, 현재 처리 영역 메모리의 출력단과 계산부(32)의 사이, 보간부(212)의 출력단과 계산부(32)의 사이, 및 이전 탐색 영역 데이터 공급부(30)와 계산부의 사이에는 스위치들이 구비된다. 본 실시예에서 상기 스위치들은 움직임 추정 제어부(270)에 의하여 제어된다.

이전 탐색 영역 데이터 공급부(30)는 제1 이전 탐색 영역 데이터 공급부(30a), 제2 이전 탐색 영역 데이터 공급부(30b), 및 제3 이전 탐색 영역 데이터 공급부(30c)를 구비한다. 계산부(32)는 제1 절대차합 계산부(322), 제2 절대차합 계산부(324), 제3 절대차합 계산부(326), 및 제4 절대차합 계산부(328)를 구비한다. 시프트레지스터부(34)는 제1 9시프트레지스터(340), 제2 9시프트레지스터(344), 제3 9시프트레지스터(348), 및 가산기들(342, 346, 350)을 구비한다.

동작을 설명하면, 부분 SAD 계산부(210)는 이전 탐색 영역 메모리와 현재 처리 영역 메모리로부터 이전 탐색 영역에 해당하는 데이터와 현재 처리 블록에 해당하는 데이터를 입력하여 부분 SAD 값을 계산한다. DMA 제어부(260)는 이전 탐색 영역에 해당하는 데이터와 현재 처리 블록에 해당하는 데이터를 구분하는 동작을 수행한다. 부분 SAD 계산부(210)는 4×4 블록에 대하여 $-4 \sim +4$ 탐색 영역까지만 연산할 수 있기 때문에 본 실시예에 따른 움직임 추정기는 가산기(220), 버퍼(222), 및 25 시프트레지스터(230)를 구비하여 4×4 블록 보다 큰 블록에 대하여 SAD 값을 얻는다.

부분 SAD 계산부(210: PE)는 이전 탐색 영역 메모리들(이전 탐색 영역 메모리_A, 이전 탐색 영역 메모리_B, 이전 탐색 영역 메모리_C)에 의한 세 채널과, 현재 처리 영역 메모리에 의한 하나의 채널을 통하여 각각 이전 탐색 영역 데이터와 현재 처리 영역 데이터를 입력받는다.

도 4에는 이전 탐색 영역의 데이터가 메모리에 배치되는 순서를 나타내었다. 도 4를 참조하면, 이전 탐색 영역의 첫 번째 라인의 데이터는 이전 탐색 영역 메모리_A, 두 번째 라인의 데이터는 이전 탐색 영역 메모리_B, 세 번째 라인의 데이터는 이전 탐색 영역 메모리_C로부터 각각 출력된다. 다음으로, 이전 탐색 영역의 네 번째 라인의 데이터는 다시 이전 탐색 영역 메모리_A로부터 출력된다. 도 4에서 이전 탐색 영역 메모리_A로부터 출력되는 데이터는 ○, 이전 탐색 영역 메모리_B로부터 출력되는 데이터는 ◇, 이전 탐색 영역 메모리_C로부터 출력되는 데이터는 □로 나타내었다.

또한, 부분 SAD 계산부(210)에는 현재 처리 영역 메모리로부터 현재 처리 영역 데이터가 입력된다. 도 5에는 4×4 픽셀로 이루어지는 현재 처리 영역의 데이터가 도 3의 계산부내의 레지스터들에 배치되는 순서를 나타내었다. 4×4 픽셀로 이루어지는 현재 처리 영역의 데이터는 도 5에 나타낸 바와 같이 도 3의 레지스터들(CR0, CR1, ..., CR15)에 로드된다. 이와같이, 현재 처리 영역 데이터들이 각 레지스터들(CR0, CR1, ..., CR15)에 저장되는 동안, 이전 탐색 영역의 데이터들도 레지스터들(PAR0, PAR1, ..., PAR4, PBR1, ..., PBR4, PCRO, ..., PCR4)에 로드된다. 레지스터들에 데이터 로드가 완료되면, 부분 SAD 계산부(210)의 차계산기(SUB)들은 이전 탐색 영역 데이터와 현재 처리 영역 데이터 사이의 차이값을 계산한다. 다음으로, 도시하지 않은 절대차 계산부는 상기 차이값을 입력하여 상기 차이값의 절대값을 출력한다. 각 절대차 계산부들의 출력은 각 픽셀들에 해당하는 연산값들이고, 이 연산값들은 현재 처리 영역 데이터의 수평 성분들끼리 더해진다. 이로써, 이전 탐색 영역 데이터가 한번 입력될 때, 현재 처리 영역의 각 라인에 대하여 차의 절대값이 계산된다.

이후의 설명을 위하여 도 4에 나타낸 이전 탐색 영역 데이터에서 최좌측 최상단의 데이터를 $p(0,0)$, $p(0,0)$ 에서 오른쪽으로 한 픽셀 만큼 시프트한 위치의 픽셀 데이터를 $p(0,1)$ 로 정의하고, $p(0,0)$ 에서 오른쪽으로 두 픽셀 만큼 시프트한 위치의 픽셀 데이터를 $p(0,2)$ 로 정의한다. 또한, $p(0,0)$ 에서 밑으로 한 픽셀 시프트한 위치의 픽셀 데이터를 $p(1,0)$, $p(0,0)$ 에서 밑으로 두 픽셀 시프트한 위치의 픽셀 데이터를 $p(2,0)$ 로 정의한다.

이와 동일하게, 도 5에 나타낸 현재 처리 영역에서 최좌측상단의 데이터를 $c(0,0)$, $c(0,0)$ 에서 오른쪽으로 한 픽셀 만큼 시프트한 위치의 픽셀 데이터를 $c(0,1)$ 로 정의하고, $c(0,0)$ 에서 오른쪽으로 두 픽셀 만큼 시프트한 위치의 픽셀 데이터를 $c(0,2)$ 로 정의한다. 또한, $c(0,0)$ 에서 밑으로 한 픽셀 시프트한 위치의 픽셀 데이터를 $c(1,0)$, $c(0,0)$ 에서 밑으로 두 픽셀 시프트한 위치의 픽셀 데이터를 $c(2,0)$ 로 정의한다.

이와 같이 정의하면, 레지스터(CR0)에는 데이터 $c(0,0)$ 가 로드되고, 레지스터(CR1)에는 데이터 $c(0,1)$ 가 로드된다. 또한, 레지스터(CR4)에는 데이터 $c(1,0)$ 가 로드되고, 레지스터(CR5)에는 데이터 $c(1,1)$ 가 로드되며, 레지스터(CR15)에는 데이터 $c(3,3)$ 가 로드된다.

부분 SAD 계산부(210)는 현재 처리 영역내에서 하나의 라인씩에 대하여 연산을 수행함으로, 제1 계산부(322), 제2 계산부(324), 제3 계산부(326), 및 제4 계산부(328)의 가산기로부터 출력되는 연산 결과값 $f(0(0,0))$, $f(1(0,0))$, $f(2(0,0))$, 및 $f(3(0,0))$ 은 각각,

$$f(0(0,0)) = c(0,0) - p(0,0) + c(0,1) - p(0,1) + c(0,2) - p(0,2) + c(0,3) - p(0,3)$$

$$f_1(0,0) = c(1,0) \cdot p(0,0) + c(1,1) \cdot p(0,1) + c(1,2) \cdot p(0,2) + c(1,3) \cdot p(0,3)$$

$$f_2(0,0) = c(2,0) \cdot p(0,0) + c(2,1) \cdot p(0,1) + c(2,2) \cdot p(0,2) + c(2,3) \cdot p(0,3)$$

$$f_3(0,0) = c(3,0) \cdot p(0,0) + c(3,1) \cdot p(0,1) + c(3,2) \cdot p(0,2) + c(3,3) \cdot p(0,3)$$

와 같다. 연산 결과값 $f_0(0,0)$, $f_1(0,0)$, $f_2(0,0)$, 및 $f_3(0,0)$ 은 현재 처리 영역 데이터의 첫 번째, 두 번째, 세 번째, 및 네 번째 라인에 대한 연산 결과에 해당한다. f 를 행, j 를 열을 나타내는 양의 수라 할 때, 이전 탐색 영역 데이터를 $P(i,j)$ 로 나타내면, 이전 탐색 영역 데이터 $P(i,j)$ 와 현재 처리 영역 데이터의 첫 번째, 두 번째, 세 번째, 및 네 번째 라인에 대한 연산 결과는 각각 $f_0(0,j)$, $f_1(0,0)$, $f_2(0,0)$, 및 $f_3(0,0)$ 로 나타내어진다. 이러한 표현에 따르면, 탐색 영역의 최좌측상단에 대한 SAD 값은,

$$f_0(0,0) + f_1(1,0) + f_2(2,0) + f_3(3,0)$$

와 같이 나타낼 수 있다. 이전 탐색 영역 데이터는 부분 SAD 계산부(210)로 한 라인씩 입력되므로, 수학식 4에서 구해지는 SAD 값이 첫 번째로 구해지는 SAD 값이 된다.

$f_0(0,0)$ 는 첫 번째 라인에 대한 연산에 의하여 구해진다. 하지만, $f_1(1,0)$ 는 이전 탐색 영역 데이터의 두 번째 라인이 입력되어야 구할 수 있고, $f_2(2,0)$ 는 세 번째 라인, $f_3(3,0)$ 는 네 번째 라인이 입력되어야 구할 수 있다. 즉, $f_0(0,0)$ 와 $f_1(1,0)$ 는 탐색 영역(search area)을 $-p \sim +p$ 로 할 때, $2p + 1$ 에 해당하는 시간 차이가 발생한다. 따라서, $f_0(0,0)$ 와 $f_1(1,0)$ 를 더하기 위해서 $2p + 1$ 클럭 만큼의 지연이 필요하다. 본 실시예에서는 탐색 영역을 $-4 \sim +4$ 로 하므로, 9 클럭 만큼의 지연이 필요하며, 시프트 레지스터부(34)의 9 시프트레지스터들(340, 344, 348)이 이러한 지연 동작을 수행한다. 다시 도 3a를 참조하여 상세히 설명하면, 제1 9시프트레지스터(340)는 제1 절대차합 계산부(322)로부터 출력된 절대차합 데이터를 9 시프트시켜 출력한다. 제1 가산기(342)는 제1 9시프트레지스터(340)의 출력 데이터와 제2 절대차합 계산부(324)의 출력 데이터를 합산한다. 제2 9시프트레지스터(344)는 제2 절대차합 계산부(324)로부터 출력된 절대차합 데이터를 9 시프트시켜 출력한다. 제2 가산기(346)는 제2 9시프트레지스터(344)의 출력 데이터와 제3 절대차합 계산부(326)의 출력 데이터를 합산한다. 제3 9시프트레지스터(348)는 제3 절대차합 계산부(326)로부터 출력된 절대차합 데이터를 9 시프트시켜 출력한다. 제3 가산기(350)는 제3 9시프트레지스터(348)의 출력 데이터와 제4 절대차합 계산부(328)의 출력 데이터를 합산한다. 제3 가산기(350)로부터 출력된 데이터는 주어진 탐색 영역에 대한 부분 SAD 값이다.

도 6을 참조하면, 본 실시예에 따른 움직임 추정기에서는 $f_0(0,0)$ 와 $f_1(1,0)$ 를 더하기 위해서 9 클럭의 지연이 필요하다. 또한, 본 실시예에 따른 움직임 추정기에서는 첫 번째 라인을 레지스터부(34)에 로딩하는데 필요한 9 클럭 만큼의 처리 지연을 제외한, 다른 처리 지연은 발생하지 않는다. 즉, 빗금 무늬의 $f_0(0,0)$, $f_1(1,0)$, $f_2(2,0)$, 및 $f_3(3,0)$ 가 매 클럭마다 순차적으로 구해지고, 다음으로 점 무늬의 $f_0(0,1)$, $f_1(1,1)$, $f_2(2,1)$, 및 $f_3(3,1)$ 가 매 클럭마다 순차적으로 구해진다. 따라서, 4×4 픽셀의 탐색 영역을 전역 탐색(full searching)에 의하여 각각의 SAD 값을 구하는데 필요한 총 시간은 초기 데이터 로딩 시간을 제외할 때 9×12 클럭, 즉, 108 클럭에 해당한다.

또한, 본 발명에 따른 움직임 추정기는 정화소 단위의 탐색을 완료하면, 정화소 탐색을 통하여 구한 움직임 벡터 지정(motion vector point)을 기준으로 반화소 단위의 탐색을 수행한다.

보간부(212)는 3 개의 이전 탐색 영역 메모리들(이전 탐색 영역 메모리_A, 이전 탐색 영역 메모리_B, 이전 탐색 영역 메모리_C)로부터 데이터를 받아들인다.

도 7을 참조하여 본 발명의 실시예에 따른 움직임 추정기에서 적용되는 반화소 단위의 탐색 과정을 설명한다. 반화소 단위의 탐색을 수행하기 위해서는 탐색점이 결정되어야 한다. 반화소 단위의 탐색점을 각각 $\alpha(i,j)$, $\beta(i,j)$, 및 $\gamma(i,j)$ 라 하면, 이들 탐색점들은 각각,

$$\alpha(i,j) = p(i,j) + p(i+1,j) + p(i,j+1) + p(i+1,j+1)/4$$

$$\beta(i,j) = p(i,j) + p(i+1,j)/2$$

$$\gamma(i,j) = p(i,j) + p(i,j+1)/2$$

과 같이 정의된다. 예를들어, $\alpha(0,0)$ 에 해당하는 탐색점은 70, $\beta(0,1)$ 에 해당하는 탐색점은 72,

$\gamma(1.0)$ 에 해당하는 탐색점은 74로 참조된다. 또한, $\alpha(0.4)$ 에 해당하는 탐색점은 76, $\gamma(1.4)$ 에 해당하는 탐색점은 78로 참조된다.

도 8에는 참고로 상기 움직임 추정기의 보간부(212)의 일예를 나타내었다. 도 8을 참조하면, 예를들어, 보간부(212)는 레지스터들(PAR0, PBR0, PCR0), 또는 레지스터들(PAR1, PBR1, PCR1), 레지스터들(PAR2, PBR2, PCR2), 레지스터들(PAR3, PBR3, PCR3), 및 레지스터들(PAR4, PBR4, PCR4) 중에서 각각 두 레지스터들로부터 이전 탐색 영역 데이터를 선택하여 입력하고, 수식식 5a, 5b, 및 5c에 나타난 등식에 따라 보간을 수행함으로써 α , β , 및 γ 픽셀값을 구하고, 스위치에 의하여 순차적으로 출력된다.

이로써, 보간부(212)는 최상위비트(MSB: most significant bit)와 최하위비트(LSB: least significant bit)를 포함하여 4 비트의 보간된 데이터를 출력한다. 계산부(32)의 입력 데이터 선택을 수행하는 스위치는 보간부(212)의 출력단자로 절환되고, 계산부(32) 및 시프트레지스터부(34)에 의하여 보간된 데이터에 대한 부분 SAD 값이 구해진다.

즉, 본 발명에 따른 움직임 추정기는 한 번 이전 프레임 영역 데이터를 데이터를 읽어들이므로써 α , β , 및 γ 에 해당하는 탐색점의 픽셀값을 동시에 생성할 수 있다. 따라서, 본 발명에 따른 움직임 추정기는 종래기술의 움직임 추정기에 비하여 처리 속도가 빠르다.

또한, 한 4×4 영역에 대한 움직임 추정이 완료되고 나서, 다음의 4×4 영역에서 움직임 추정을 하는 경우, 예를들어, $\alpha(0.4)$ 에 해당하는 탐색점(76), $\beta(1.4)$ 에 해당하는 탐색점(78)의 픽셀값은 이미 읽은 이전 탐색 영역 데이터를 시프트 라이트 함으로써 구할 수 있다. 따라서, 본 발명에 따른 움직임 추정기는 특히 반화소 단위의 탐색을 빠르고 효율적으로 수행할 수 있다.

도 9에는 도 2의 나타난 움직임 추정기에 반화소 단위의 탐색을 적용하였을 경우의 데이터 흐름을 나타내었다. 도 9를 참조하면, 상기 움직임 추정기에서는 하나의 4×4 블록에 대한 반화소 단위의 탐색이 24 클럭 동안에 완료된다.

한편, 본 실시예에 따른 움직임 추정기에 구비된 부분 SAD 계산부(210)의 처리 단위는 4×4 블록이다. 이러한 움직임 추정기를 3 계층(layer)의 계층적 탐색(hierarchical search)에 사용할 때, 상위 계층의 경우는 4×4 블록내에서 -4 ~ +4 픽셀의 탐색 영역에 대하여 전역 탐색을 수행하고, 중간 계층의 경우는 8×8 블록내에서 -2 ~ +2 픽셀의 탐색 영역에 대하여 탐색을 수행하며, 하위 계층의 경우는 16×16 블록내에서 -2 ~ +2 픽셀의 탐색 영역에 대하여 탐색을 수행한다.

본 실시예에 따른 움직임 추정기의 부분 SAD 계산부(210)는 4×4 블록을 계산할 수 있다. 따라서, 8×8 블록의 중간계층과 16×16 블록의 하위 계층에 대하여 움직임 추정을 수행하기 위해서는 계층들을 각각 4 개 및 16 개의 블록들로 나누어 계산한 다음, 결과값들을 더해주는 것이 필요하다. 하지만, 중간 계층 및 하위 계층에서는 탐색 영역이 도 10에 도시한 바와 같이 -2 ~ +2 픽셀에 해당하므로, 한 번 탐색을 수행하면 25 포인트에 대한 부분 SAD 값을 구할 수 있다. 이를 위하여, 도 2를 참조하면, 본 발명에 따른 움직임 추정기는 25 시프트레지스터(230)를 구비한다. 상기 움직임 추정기는 부분 SAD 값을 25 시프트레지스터(230)에 입력시키고, 다음의 부분 SAD 값이 출력될 때마다 25 시프트레지스터(230)에 저장된 데이터를 시프트시켜, 다음의 4×4 블록의 부분 SAD 값과 더함으로써 중간 계층 및 하위 계층의 탐색 작업을 수행하는 것이 가능하다.

한편, 본 실시예에 따른 움직임 추정기는 3 개의 초기 탐색점중 하나를 주변 매크로 블록과의 상관관계를 사용하여 선택한다. 주변 매크로 블록의 위치는 도 11에 도시한 바와 같이 정할 수 있다. 입력 영상 프레임의 좌상단 왼쪽으로부터 수평 방향으로 움직임 벡터를 구하기 때문에, 현재 처리 블록과 현재 처리 블록의 위에 위치한 매크로 블록 사이의 거리는 입력 영상 프레임의 수평 방향 매크로 블록 수와 같다. 따라서, 입력 영상 프레임이 352×288 화소의 CIF 형식의 영상인 것을 가정하면 입력 영상 프레임의 최대 크기는 수평 방향으로 22 개의 매크로 블록에 해당한다. 따라서, 본 발명의 타실시예에 따르면, 22 시프트 레지스터 어레이(1100) 및 중간값 계산부(1102)을 더 구비한다. 22 시프트 레지스터 어레이(1100)는 주변 매크로 블록들에 대한 움직임 벡터값을 입력하고, 중간값 계산부(1102)는 이들의 중간값을 움직임 벡터 후보로써 출력함으로써, 주변 매크로 블록과의 상관 관계를 이용하여 움직임 벡터 후보를 더 선정한다. 선정된 움직임 벡터 후보는 중간 계층의 초기 탐색점으로써 사용된다.

본 발명의 실시예에 따른 움직임 추정기에 적용되는 3 계층의 계층적 탐색에서 상위 계층 데이터는 원 데이터를 4:1 데시메이션(decimation)한 값들로 이루어진다. 본 실시예에서, 데시메이션은 원 데이터를 매 4 샘플마다 하나씩 취하는 방법을 사용한다. 현재 처리 영역의 데이터는 하나의 현재 처리 영역 메모리에 저장되어 있으므로, 수평 및 수직 방향으로 매 4 샘플마다 하나씩 취한다. 이전 탐색 영역의 데이터는 세 이전 탐색 영역 메모리에 나뉘어 저장되어 있기 때문에 수평 방향의 데시메이션은 현재 처리 영역의 데이터의 경우와 동일하게 매 4 샘플마다 하나씩 취하지만, 수직 방향의 데시메이션은 도 12에 도시한 바와 같이 이루어진다. 즉, 본 실시예에 따르면, 이전 탐색 영역 메모리는 탐색 영역 데이터의 첫 번째 라인이 이전 탐색 영역 메모리_A, 두 번째 라인이 이전 탐색 영역 메모리_B, 세 번째 라인이 이전 탐색 영역 메모리_C에 저장하는 순서에 따라 저장되어 있으므로, 도 2의 DMA 제어부(260)는 어드레스를 4 씩 증가시키고, 이전 탐색 영역 메모리_A, 이전 탐색 영역 메모리_B, 및 이전 탐색 영역 메모리_C의 순으로 읽어 내도록 제어할 것이 요구된다.

메모리로부터 읽은 데이터는 부분 SAD 계산부(210)의 각 레지스터에 저장되고 -4 ~ +4 블록의 탐색 영역에 대하여 전역 탐색을 수행함으로써 최소의 SAD 값을 가지는 두 위치를 중간 계층의 초기 탐색점으로써 선택한다. 상위 계층에서의 데이터 처리 흐름은 도 13에 나타난 바와 같다. 도 13은 기본적으로 도 6과 동일하므로 더이상 설명하지 않는다.

중간 계층의 데이터는 하위 계층의 데이터를 2:1 데시메이션함으로써 얻어진다. 따라서, 현재 처리 영역 메모리와 이전 탐색 영역 메모리로부터 수평 및 수직 방향으로 2 씩 어드레스를 증가시킴으로써 도 14에

도시한 바와 같이 메모리의 읽음 및 건너뛸을 수행한다. 이전 탐색 영역 데이터는 3 개의 메모리들에 나눠 저장되어 있으므로 수직 방향으로는 이전 탐색 영역 메모리_A, 이전 탐색 영역 메모리_B, 이전 탐색 영역 메모리_C의 순서로 데이터를 읽어온다. 이로써, 이전 탐색 영역의 데이터에 대한 수직 방향으로의 2:1 데시메이션(decimation)이 이루어진다.

중간 계층에서는 상위 계층에서 찾은 2 개의 초기 탐색점과 22 시프트레지스터(250)에서 얻어지는 움직임 벡터 후보에 해당하는 1 개의 초기 탐색점을 중심점으로 하여 결정되는 8×8 블록에 대하여 -2 ~ +2 탐색 영역의 전역 탐색을 수행하여 부분 SAD 값을 구한다. 부분 SAD 계산부(210)는 4×4 블록을 단위로 탐색을 수행하도록 설계되어 있으므로, 8×8 블록의 중간 계층에 대하여 탐색을 수행하기 위해서는 도 15에 나타난 바와 같이 현재 프레임의 8×8 블록을 4 개의 4×4 블록으로 나누어 탐색을 수행한다. 즉, 처음 4×4 블록에 대하여 -2 ~ +2 탐색 영역에 걸쳐 25 포인트의 부분 SAD 값을 구하여 상기 부분 SAD 값을 25 시프트 레지스터(230)에 저장한다. 이제, 가산기(220)에 의하여 상기 부분 SAD 값과 다음 4×4 블록에 대한 부분 SAD 값과 더한다. 이러한 과정을 4 번 반복함으로써 8×8 블록에 대한 SAD 값을 구한다. 이렇게 3 개의 초기 탐색점들에 대한 -2 ~ +2 탐색 영역에 의하여 최소의 SAD 값을 가지는 위치를 하위 계층의 초기 탐색점으로 사용한다. 중간 계층에서의 데이터 처리 흐름은 도 16에 나타난 바와 같다. 도 16을 참조하면, 중간 계층에서는 $f(0,0)$ 와 $f(1,0)$ 를 더하기 위해서 5 클럭의 지연이 필요하다. 즉, 빗금 무늬의 $f(0,0)$, $f(1,0)$, $f(2,0)$, 및 $f(3,0)$ 가 매 클럭마다 순차적으로 구해지고, 다음으로 점 무늬의 $f(0,1)$, $f(1,1)$, $f(2,1)$, 및 $f(3,1)$ 가 매 클럭마다 순차적으로 구해진다.

하위계층의 데이터는 원 데이터로 구성된다. 중간 계층에서 찾은 초기 탐색점을 중심점으로 하여 16×16 블록에 대하여 -2 ~ +2 탐색 영역에서 걸쳐 전역 탐색을 수행하여 최소의 SAD 값을 가지는 위치를 구한다. 상기 위치로부터 정화소 단위의 탐색에 의한 최종 움직임 벡터가 구해진다. 하위계층에 대한 움직임 추정은 도 17에 도시한 바와 같이 16×16 블록에 대하여 수행되고, 부분 SAD 계산부(210)는 4×4 블록을 기준으로 처리를 수행하므로, 4×4 블록 단위의 탐색을 16 번 반복함으로써 16×16 블록에 대한 SAD 값을 구한다. 하위 계층에서의 데이터 처리 흐름은 도 18에 나타난 바와 같다. 도 18에 나타난 하위 계층에서의 데이터 처리 흐름은 ± 2 픽셀의 탐색을 수행되는 중간 계층에서의 데이터 흐름을 나타낸 도 16과 동일하므로 더이상 설명하지 않는다.

발명의 효과

상술한 바와 같이, 본 발명에 따른 움직임 추정기는 처리 속도가 빠르고, 특히, 반화소 단위의 탐색을 빠르고 효율적으로 수행할 수 있다.

(57) 청구의 범위

청구항 1

영상 데이터 프레임을 압축하기 위하여 움직임 추정을 수행하는 움직임 추정기에 있어서,

이전 탐색 영역의 데이터를 라인별로 저장하는 복수 개의 메모리 영역을 구비하는 이전 탐색 영역 메모리와, 현재 처리 영역의 데이터를 라인별로 저장하는 현재 처리 영역 메모리를 구비하는 제1 메모리부;

복수 개의 메모리 영역으로부터 순환적으로 출력되는 이전 탐색 영역의 데이터를 시프트시켜 출력하는 이전 탐색 영역 데이터 공급부;

이전 탐색 영역 데이터 공급부로부터 출력된 이전 탐색 영역의 데이터를 선택하는 제1 스위칭부;

현재 처리 영역 메모리로부터 출력되는 현재 처리 영역의 데이터를 행별로 선택하는 제2 스위칭부;

선택된 이전 처리 영역의 데이터와 현재 처리 영역 데이터의 절대차합(sum of absolute difference)을 계산하는 복수 개의 절대차합 계산부가 병렬로 접속된 계산부; 및

계산부의 일단의 절대차합 계산부들로부터 출력된 절대차합 데이터를 입력하여 시프트를 수행하고 시프트된 데이터와 다음 단의 절대차합 계산부로부터 출력된 SAD(절대차합) 값을 합하여 부분 SAD 값을 출력하는 시프트 레지스터부;를 포함하는 것을 특징으로 하는 움직임 추정기.

청구항 2

제1항에 있어서,

상기 계산부는 탐색 영역의 크기를 p라 할 때 p 개의 절대차합 계산부가 병렬로 접속되고, 상기 시프트 레지스터부는 $2p + 1$ 시프트레지스터를 구비하는 것을 특징으로 하는 움직임 추정기.

청구항 3

제1항 또는 제2항 중 어느 한 항에 있어서,

입력 데이터를 버퍼링하는 버퍼;

버퍼로부터 출력된 데이터를 25 시프트하는 25 시프트레지스터;

부분 SAD 값과 25 시프트 레지스터로부터 출력된 데이터를 가산하여 SAD 값을 출력하는 가산기;

가산기로부터 출력된 SAD 값을 사용하여 움직임 벡터를 생성하는 SAD 비교 및 움직임 벡터 생성부; 및

움직임 벡터값을 20 시프트시켜 SAD 비교 및 움직임 벡터 생성부로 출력하는 20 시프트레지스터;를 더 포함하는 것을 특징으로 하는 움직임 추정기.

청구항 4

제1항에 있어서,

3 계층으로 계층적 움직임 추정을 수행할 때 중간 계층에서 주변 매크로 블록이 가지는 움직임 벡터의 상관관계를 이용하여 초기 탐색점을 추가로 얻기 위하여 22 시프트 동작을 수행하는 22 시프트레지스터;를 더 포함하는 것을 특징으로 하는 움직임 추정기.

청구항 5

제1항에 있어서,

움직임 추정을 수행하기 위하여 현재 데이터와 이전 탐색 영역 데이터를 구분하여 제1 메모리에 저장되도록 제어하고, 움직임 추정 제어부의 처리 상황에 따라 부분 SAD 계산부로 데이터를 공급하도록 제어하는 DMA 제어부;

DMA 제어부로부터 출력된 제어신호에 응답하여 움직임 추정 동작을 제어하는 움직임 추정 제어부; 및 계산된 부분 SAD 값을 저장하는 제2 메모리부;를 더 포함하는 것을 특징으로 하는 움직임 추정기.

청구항 6

제1항에 있어서,

상기 제1 메모리는,

세 영역의 이전 탐색 영역 메모리와, 현재 처리 영역 메모리로 구분되고,

상기 계산부는,

이전 탐색 영역의 데이터와 현재 처리 영역의 데이터를 사용하여 절대차합을 계산하는 제1 절대차합 계산부; 제2 절대차합 계산부; 제3 절대차합 계산부; 및 제4 절대차합 계산부;를 구비하고,

상기 시프트 레지스터부는,

상기 제1 절대차합 계산부로부터 출력된 절대차합 데이터를 9 시프트시켜 출력하는 제1 9시프트레지스터; 상기 제1 9시프트레지스터의 출력 데이터와 상기 제2 절대차합 계산부의 출력 데이터를 합산하는 제1 가산기; 상기 제2 절대차합 계산부로부터 출력된 절대차합 데이터를 9 시프트시켜 출력하는 제2 9시프트레지스터; 상기 제2 9시프트레지스터의 출력 데이터와 상기 제3 절대차합 계산부의 출력 데이터를 합산하는 제2 가산기; 제3 계산부로부터 출력된 절대차합 데이터를 9 시프트시켜 출력하는 제3 9시프트레지스터; 상기 제3 9시프트레지스터의 출력 데이터와 상기 제4 절대차합 계산부의 출력 데이터를 합산하는 제3 가산기;를 구비하는 것을 특징으로 하는 움직임 추정기.

청구항 7

제6항에 있어서,

상기 제1 절대차합 계산부는 레지스터들(CR0, CR1, CR2, CR3)과, 일 입력단은 상기 레지스터들(CR0, CR1, CR2, CR3)의 출력단에 접속되고 타 입력단은 스위치들에 접속된 감산기들(SUB)을 구비하고, 상기 제2 절대차합 계산부는 레지스터들(CR4, CR5, CR6, CR7)과, 일 입력단은 상기 레지스터들(CR4, CR5, CR6, CR7)의 출력단에 접속되고 타 입력단은 스위치들에 접속된 감산기들(SUB)을 구비하며, 상기 제3 절대차합 계산부는 레지스터들(CR8, CR9, CR10, CR11)과, 일 입력단은 상기 레지스터들(CR8, CR9, CR10, CR11)의 출력단에 접속되고 타 입력단은 스위치들에 접속된 감산기들(SUB)을 구비하고, 상기 제4 절대차합 계산부는 레지스터들(CR12, CR13, CR14, CR15)과, 일 입력단은 상기 레지스터들(CR12, CR13, CR14, CR15)의 출력단에 접속되고 타 입력단은 스위치들에 접속된 감산기들(SUB)을 구비하는 것을 특징으로 하는 움직임 추정기.

청구항 8

제1항에 있어서,

이전 탐색 영역의 데이터를 보관하여 보관된 데이터를 출력하는 보관부;를 더 포함하는 것을 특징으로 하는 움직임 추정기.

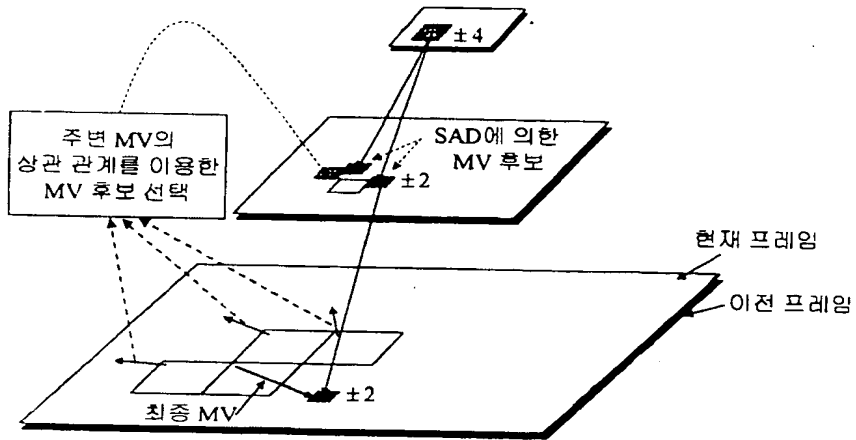
청구항 9

제1항에 있어서,

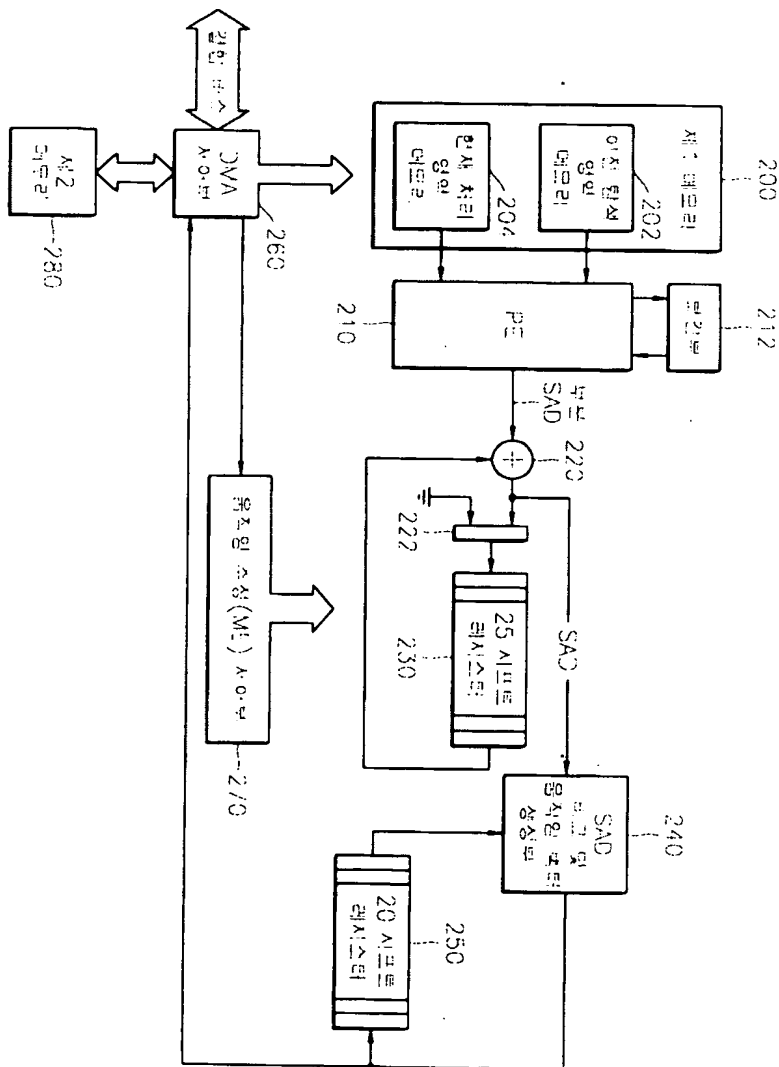
이전 탐색 영역에 해당하는 데이터와 현재 처리 블록에 해당하는 데이터를 구분하는 동작을 수행하고, 처리 상황에 따라 상기 이전 탐색 영역 데이터 공급부와 상기 계산부로 데이터를 공급하도록 제어하는 DMA 제어부를 더 포함하는 것을 특징으로 하는 움직임 추정기.

도면

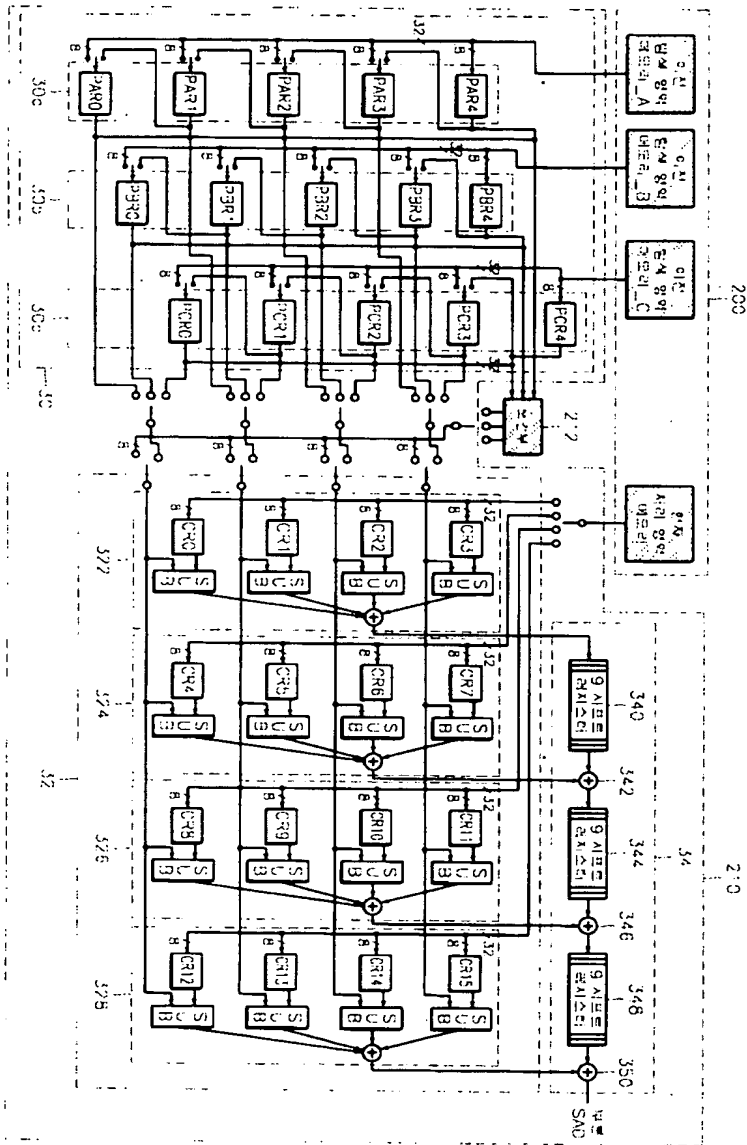
도면1



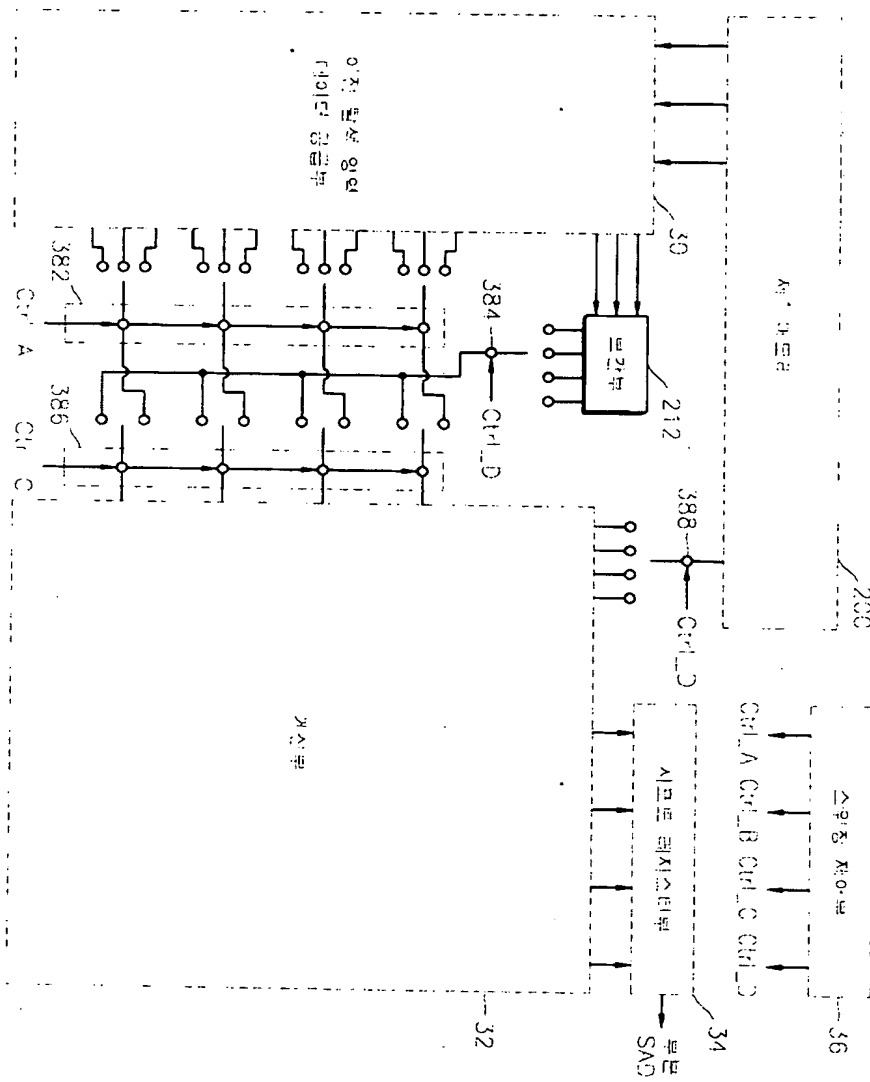
도면2



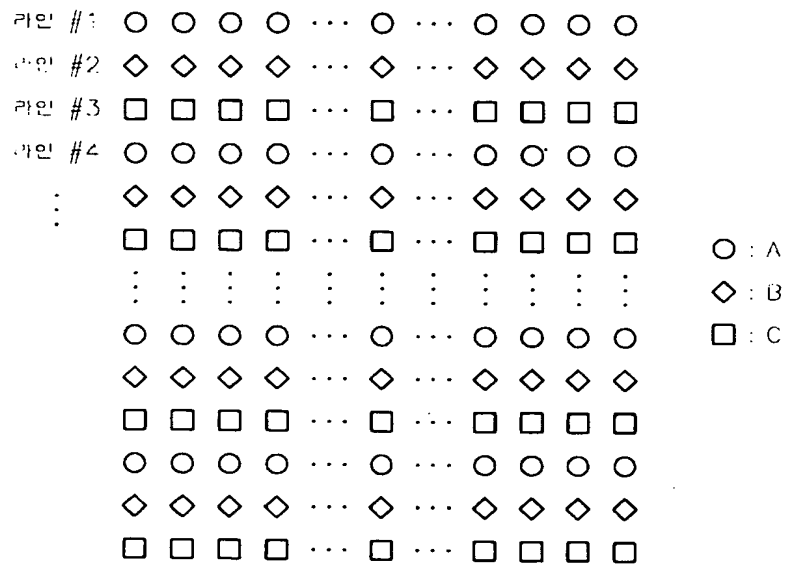
도면3a



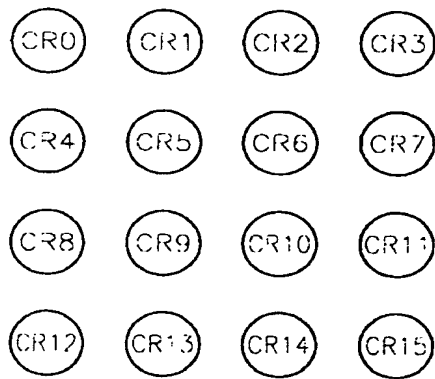
도면 3b



도면4



도면5

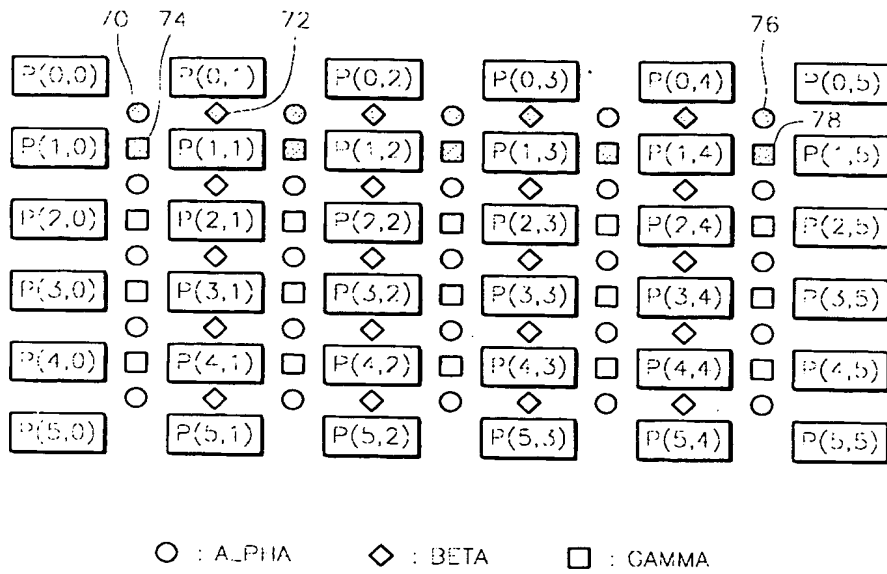


도면6

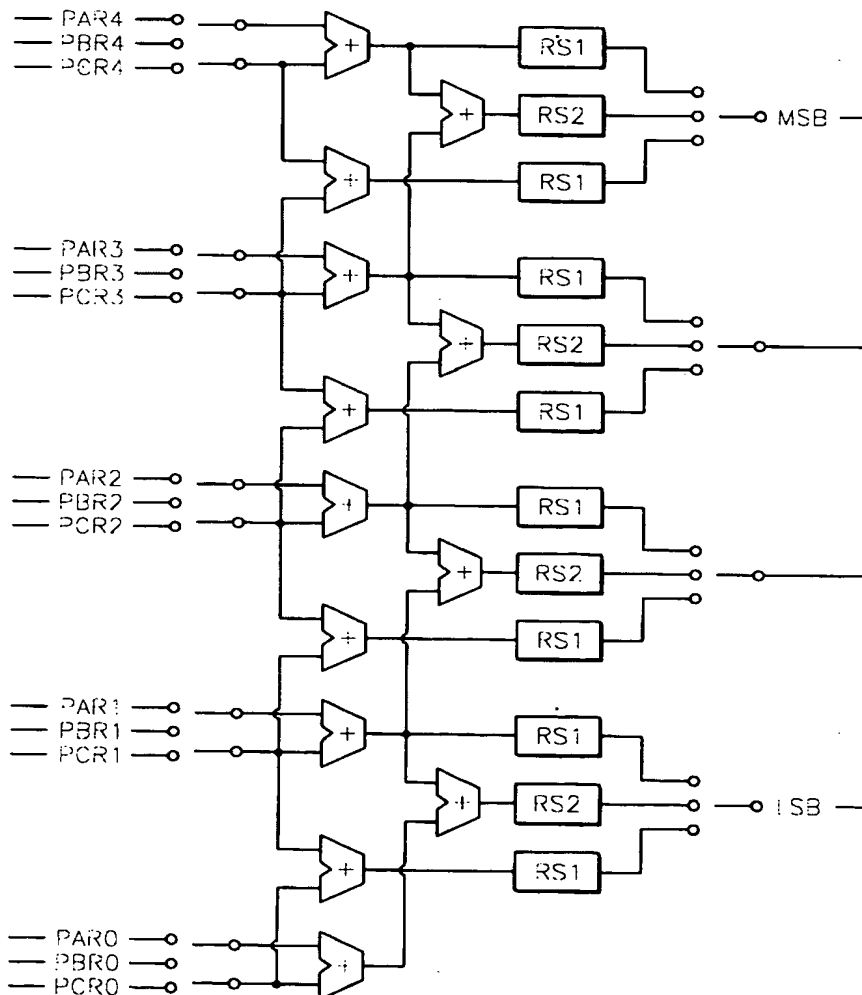
PAR#	PBR#	PCR#	L0	L1	L2	L3
PER#LOAD	PER#LOAD	PER#LOAD	LOAD	X	X	X
HOLD	HOLD	HOLD	LO(0,0)	LOAD	X	X
SHIFT	HOLD	HOLD	LO(0,1)	X	X	X
SHIFT	HOLD	HOLD	LO(0,2)	X	X	X
SHIFT	HOLD	HOLD	LO(0,3)	X	X	X
SHIFT	HOLD	HOLD	LO(0,4)	X	X	X
SHIFT	HOLD	HOLD	LO(0,5)	X	X	X
SHIFT	HOLD	HOLD	LO(0,6)	X	X	X
SHIFT	HOLD	HOLD	LO(0,7)	X	X	X
SHIFT	HOLD	HOLD	LO(0,8)	X	X	X
NEXT LINE LOAD	HOLD	HOLD	LO(1,0)	L1(1,0)	LOAD	X
HOLD	SHIFT	HOLD	LO(1,1)	L1(1,1)	X	X
HOLD	SHIFT	HOLD	LO(1,2)	L1(1,2)	X	X
:			:			
HOLD	SHIFT	HOLD	LO(1,8)	L1(1,8)	X	X
HOLD	NEXT LINE LOAD	HOLD	LO(2,0)	L1(2,0)	L2(2,0)	LOAD
HOLD	HOLD	SHIFT	LO(2,1)	L1(2,1)	L2(2,1)	X
HOLD	HOLD	SHIFT	LO(2,2)	L1(2,2)	L2(2,2)	X
:			:			
HOLD	HOLD	SHIFT	LO(2,8)	L1(2,8)	L2(2,8)	X
HOLD	HOLD	NEXT LINE LOAD	LO(3,0)	L1(3,0)	L2(3,0)	L3(3,0)
SHIFT	HOLD	HOLD	LO(3,1)	L1(3,1)	L2(3,1)	L3(3,1)
SHIFT	HOLD	HOLD	LO(3,2)	L1(3,2)	L2(3,2)	L3(3,2)
:			:			
SHIFT	HOLD	HOLD	LO(3,8)	L1(3,8)	L2(3,8)	L3(3,8)
NEXT LINE LOAD	HOLD	HOLD	LO(4,0)	L1(4,0)	L2(4,0)	L3(4,0)
HOLD	SHIFT	HOLD	LO(4,1)	L1(4,1)	L2(4,1)	L3(4,1)
HOLD	SHIFT	HOLD	LO(4,2)	L1(4,2)	L2(4,2)	L3(4,2)
:			:			

SHIFT

도면7



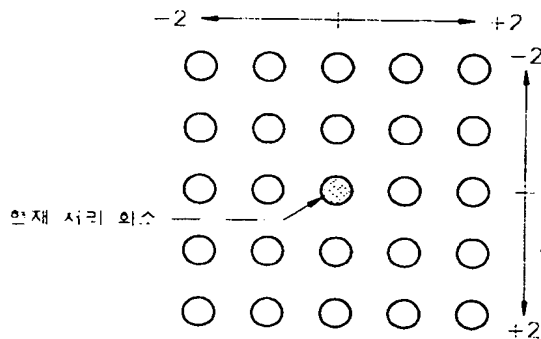
도면8



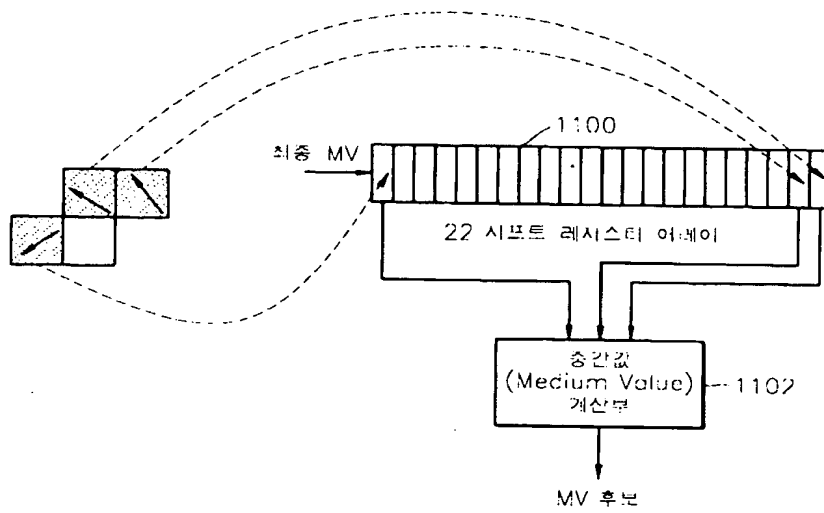
도면9

PAR#	PBR#	PCR#	L0	L1	L2	L3
PER#LOAD	PER#LOAD	POR#LOAD	LOAD	X	X	X
HOLD	HOLD	HOLD	ALPHA LO(0,0)	LOAD	X	X
HOLD	HOLD	HOLD	BETA LO(0,0)	X	X	X
HOLD	HOLD	HOLD	GAMMA LO(0,0)	X	X	X
SHIFT	SHIFT	HOLD	ALPHA LO(0,1)	X	X	X
HOLD	HOLD	HOLD	GAMMA LO(0,1)	X	X	X
NEXT LINE LOAD	RFLOAD	HOLD	ALPHA LO(1,0)	ALPHA L1(1,0)	LOAD	X
HOLD	HOLD	HOLD	BETA LO(1,0)	BETA L1(1,0)	X	X
HOLD	HOLD	HOLD	GAMMA LO(1,0)	GAMMA L1(1,0)	X	X
HOLD	SHIFT	SHIFT	ALPHA LO(1,1)	ALPHA L1(1,1)	X	X
HOLD	HOLD	HOLD	GAMMA LO(1,1)	GAMMA L1(1,1)	X	X
HOLD	NEXT LINE LOAD	RELOAD	ALPHA LO(1,0)	ALPHA L1(2,0)	ALPHA L2(2,0)	LOAD
HOLD	HOLD	HOLD	BETA LO(2,0)	BETA L1(2,0)	BETA L2(2,0)	X
HOLD	HOLD	HOLD	GAMMA LO(2,0)	GAMMA L1(2,0)	GAMMA L2(2,0)	X
SHIFT	HOLD	SHIFT	ALPHA LO(2,1)	ALPHA L1(2,1)	ALPHA L2(2,1)	X
HOLD	HOLD	HOLD	GAMMA LO(2,1)	GAMMA L1(2,1)	GAMMA L2(2,1)	X
RELOAD	RELOAD	NEXT LINE LOAD	ALPHA LO(3,0)	ALPHA L1(3,0)	ALPHA L2(3,0)	ALPHA L3(3,0)
HOLD	HOLD	HOLD	BETA LO(3,0)	BETA L1(3,0)	BETA L2(3,0)	BETA L3(3,0)
HOLD	HOLD	HOLD	GAMMA LO(3,0)	GAMMA L1(3,0)	GAMMA L2(3,0)	GAMMA L3(3,0)
SHIFT	SHIFT	HOLD	ALPHA LO(3,1)	ALPHA L1(3,1)	ALPHA L2(3,1)	ALPHA L3(3,1)
HOLD	HOLD	HOLD	GAMMA LO(3,1)	GAMMA L1(3,1)	GAMMA L2(3,1)	GAMMA L3(3,1)
X	RFLOAD	HOLD	ALPHA LO(4,0)	ALPHA L1(4,0)	ALPHA L2(4,0)	ALPHA L3(4,0)
X	HOLD	HOLD	BETA LO(4,0)	BETA L1(4,0)	BETA L2(4,0)	BETA L3(4,0)
X	HOLD	HOLD	GAMMA LO(4,0)	GAMMA L1(4,0)	GAMMA L2(4,0)	GAMMA L3(4,0)
X	SHIFT	SHIFT	ALPHA LO(4,1)	ALPHA L1(4,1)	ALPHA L2(4,1)	ALPHA L3(4,1)

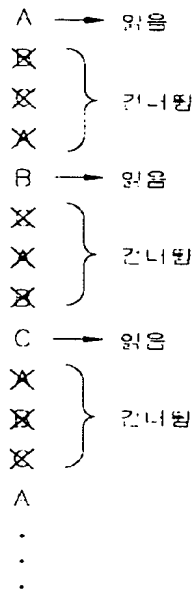
도면10



도면11



도면12



도면 13

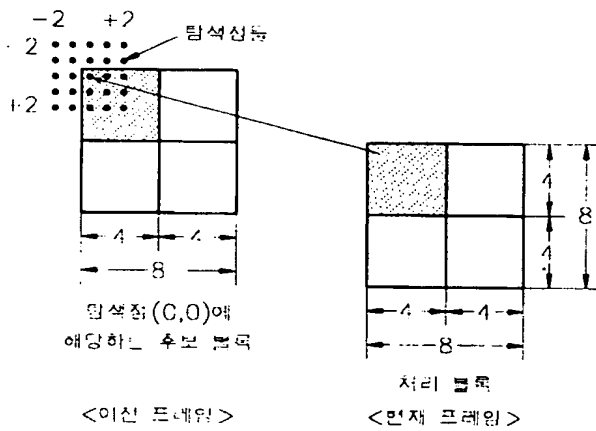
PAR#	PR#	PCR#	LO	L1	L2	L3
1 st LINE LOAD	2 nd LINE LOAD	3 rd LINE LOAD	LOAD	X	X	X
HOLD	HOLD	HOLD	LO(0,0)	LOAD	X	X
SHIFT	HOLD	HOLD	LO(0,1)	X	X	X
SHIFT	HOLD	HOLD	LO(0,2)	X	X	X
SHIFT	HOLD	HOLD	LO(0,3)	X	X	X
SHIFT	HOLD	HOLD	LO(0,4)	X	X	X
SHIFT	HOLD	HOLD	LO(0,5)	X	X	X
SHIFT	HOLD	HOLD	LO(0,6)	X	X	X
SHIFT	HOLD	HOLD	LO(0,7)	X	X	X
SHIFT	HOLD	HOLD	LO(0,8)	X	X	X
5 th LINE LOAD	HOLD	HOLD	LO(1,0)	L1(1,0)	LOAD	X
HOLD	SHIFT	HOLD	LO(1,1)	L1(1,1)	X	X
HOLD	SHIFT	HOLD	LO(1,2)	L1(1,2)	X	X
⋮			⋮			
HOLD	SHIFT	HOLD	LO(1,8)	L1(1,8)	X	X
HOLD	6 th LINE LOAD	HOLD	LO(2,0)	L1(2,0)	L2(2,0)	LOAD
HOLD	HOLD	SHIFT	LO(2,1)	L1(2,1)	L2(2,1)	X
HOLD	HOLD	SHIFT	LO(2,2)	L1(2,2)	L2(2,2)	X
⋮			⋮			
HOLD	HOLD	SHIFT	LO(2,8)	L1(2,8)	L2(2,8)	X
HOLD	HOLD	7 th LINE LOAD	LO(3,0)	L1(3,0)	L2(3,0)	L3(3,0)
SHIFT	HOLD	HOLD	LO(3,1)	L1(3,1)	L2(3,1)	L3(3,1)
SHIFT	HOLD	HOLD	LO(3,2)	L1(3,2)	L2(3,2)	L3(3,2)
⋮			⋮			
SHIFT	HOLD	HOLD	LO(3,8)	L1(3,8)	L2(3,8)	L3(3,8)
9 th LINE LOAD	HOLD	HOLD	LO(4,0)	L1(4,0)	L2(4,0)	L3(4,0)
HOLD	SHIFT	HOLD	LO(4,1)	L1(4,1)	L2(4,1)	L3(4,1)
HOLD	SHIFT	HOLD	LO(4,2)	L1(4,2)	L2(4,2)	L3(4,2)
⋮			⋮			

시리들 막히어 선택된 데이터

도면14


A → 위쪽
 X → 건너뛰
 C → 위쪽
 X → 건너뛰
 B
 X
 A
 X
 C
 X
 B
 X
 A
 :
 :

도면15



도면 16


PAR#	PBR#	PCR#	L0	L1	L2	L3
1st LINE LOAD	2nd LINE LOAD	1st LINE LOAD	LOAD	X	X	X
HOLD	HOLD	HOLD	LO(0,0)	LOAD	X	X
SHIFT	HOLD	HOLD	LO(0,1)	X	X	X
SHIFT	HOLD	HOLD	LO(0,2)	X	X	X
SHIFT	HOLD	HOLD	LO(0,3)	X	X	X
SHIFT	HOLD	HOLD	LO(0,4)	X	X	X
3rd LINE LOAD	HOLD	HOLD	LO(1,0)	L1(1,0)	LOAD	X
HOLD	HOLD	SHIFT	LO(1,1)	L1(1,1)	X	X
HOLD	HOLD	SHIFT	LO(1,2)	L1(1,2)	X	X
HOLD	HOLD	SHIFT	LO(1,3)	L1(1,3)	X	X
HOLD	HOLD	SHIFT	LO(1,4)	L1(1,4)	X	X
HOLD	HOLD	3rd LINE LOAD	LO(2,0)	L1(2,0)	L2(2,0)	X
HOLD	SHIFT	HOLD	LO(2,1)	L1(2,1)	L2(2,1)	X
HOLD	SHIFT	HOLD	LO(2,2)	L1(2,2)	L2(2,2)	X
HOLD	SHIFT	HOLD	LO(2,3)	L1(2,3)	L2(2,3)	X
HOLD	SHIFT	HOLD	LO(2,4)	L1(2,4)	L2(2,4)	X
HOLD	4th LINE LOAD	HOLD	LO(3,0)	L1(3,0)	L2(2,0)	L3(3,0)
SHIFT	HOLD	HOLD	LO(3,1)	L1(3,1)	L2(2,1)	L3(3,1)
SHIFT	HOLD	HOLD	LO(3,2)	L1(3,2)	L2(2,2)	L3(3,2)
SHIFT	HOLD	HOLD	LO(3,3)	L1(2,3)	L2(2,3)	L3(3,3)
SHIFT	HOLD	HOLD	LO(3,4)	L1(2,4)	L2(2,4)	L3(3,4)
5th LINE LOAD	HOLD	HOLD	LO(4,0)	L1(4,0)	L2(4,0)	L3(4,0)
HOLD	HOLD	SHIFT	LO(4,1)	L1(4,1)	L2(4,1)	L3(4,1)
HOLD	HOLD	SHIFT	LO(4,2)	L1(4,2)	L2(4,2)	L3(4,2)
			⋮	⋮		

 : 시리클을 위하여 선택된 데이터

↑
↓
시리클

도면18

PAR#	PBR#	PCR#	LO	L1	L2	L3
1 st LINE LOAD	2 nd LINE LOAD	1 st LINE LOAD	LOAD	X	X	X
HOLD	HOLD	HOLD	LO(0,0)	LOAD	X	X
SHIFT	HOLD	HOLD	LO(0,1)	X	X	X
SHIFT	HOLD	HOLD	LO(0,2)	X	X	X
SHIFT	HOLD	HOLD	LO(0,3)	X	X	X
SHIFT	HOLD	HOLD	LO(0,4)	X	X	X
2 nd LINE LOAD	HOLD	HOLD	LO(1,0)	L1(1,0)	LOAD	X
HOLD	SHIFT	HOLD	LO(1,1)	L1(1,1)	X	X
HOLD	SHIFT	HOLD	LO(1,2)	L1(1,2)	X	X
HOLD	SHIFT	HOLD	LO(1,3)	L1(1,3)	X	X
HOLD	SHIFT	HOLD	LO(1,4)	L1(1,4)	X	X
HOLD	2 nd LINE LOAD	HOLD	LO(2,0)	L1(2,0)	L2(2,0)	X
HOLD	HOLD	SHIFT	LO(2,1)	L1(2,1)	L2(2,1)	X
HOLD	HOLD	SHIFT	LO(2,2)	L1(2,2)	L2(2,2)	X
HOLD	HOLD	SHIFT	LO(2,3)	L1(2,3)	L2(2,3)	X
HOLD	HOLD	SHIFT	LO(2,4)	L1(2,4)	L2(2,4)	X
HOLD	HOLD	2 nd LINE LOAD	LO(3,0)	L1(3,0)	L2(2,0)	L3(3,0)
SHIFT	HOLD	HOLD	LO(3,1)	L1(3,1)	L2(2,1)	L3(3,1)
SHIFT	HOLD	HOLD	LO(3,2)	L1(3,2)	L2(2,2)	L3(3,2)
SHIFT	HOLD	HOLD	LO(3,3)	L1(2,3)	L2(2,3)	L3(3,3)
SHIFT	HOLD	HOLD	LO(3,4)	L1(2,4)	L2(2,4)	L3(3,4)
3 rd LINE LOAD	HOLD	HOLD	LO(4,0)	L1(4,0)	L2(4,0)	L3(4,0)
HOLD	SHIFT	HOLD	LO(4,1)	L1(4,1)	L2(4,1)	L3(4,1)
HOLD	SHIFT	HOLD	LO(4,2)	L1(4,2)	L2(4,2)	L3(4,2)
			⋮	⋮		

 : 처리를 위하여 선택된 데이터

5
단
위